2003

Computer assisted eye fungal infection diagnosis

Madhusudhanan Balasubramanian
Louisiana State University and Agricultural and Mechanical College, mbalas1@lsu.edu

COMPUTER ASSISTED EYE FUNGAL INFECTION DIAGNOSIS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agriculture and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Systems Science

in

The Department of Computer Science

by
Madhusudhanan Balasubramanian
B.E., Bharathiyar University, 1998
December 2003

Dedicated to my parents,
Sudha and Balu

Acknowledgments

I would like to thank my committee chair and my major professor, Dr. Louise A. Perkins, for her supervision, constructive criticism, valuable guidance, several long passionate discussions and enthusiasm, she provided all through the course of this thesis and several other research work. I sincerely thank her for allowing me to try and work on versatile research ideas, which definitely improved my research quality.

I thank Dr. S. Sitharama Iyengar for several discussions and insightful ideas with regard to this work. Also, I would like to thank Dr. Donald H. Kraft for the guidance and for being part of this thesis committee. I would like to thank Dr. John Tyler for several discussions on this work.

I would like to thank Dr. Roger Beuerman, professor, LSU School of Medicine for the several discussions on this work and for providing the images for this work. Most of all, I would like to thank my parents whose generous sacrifices and care made this work possible. I thank the god for being with me all the time.

Table of Contents

List of Tables

List of Figures

Abstract

The eye, a vital organ of the human body, gives us the sense of color, shape and state of physical objects. Fungal Keratitis, a fungal infection that occurs in the corneal layers of the eye, is one of the major causes of blindness. Hence it becomes important to diagnose and treat the fungal infection at the earliest because of the devastating ocular damage it can cause to the eye.

In this thesis, an attempt has been made to assist the diagnosis of Fungal Keratitis by identifying the region of infection in the corneal images using fractal-based features. Three features related to the fractal dimension of the surface of the image, when represented in a 3D using the pixel intensity measure, are used to identify these regions in the image.

To reduce the computation complexity, Fisher linear discriminant (FLD) is used to reduce the 3D raw feature to 1D feature, while preserving feature values. Using the adaptive mixtures (AM) method, the probability density distribution of the two class fractal features, is estimated. A training corneal image has been used to build the two-class probability density distribution.

In this work, we use Bayesian classifier, a standard statistical pattern classification technique, to classify the pixels in corneal images, using the two-class probability density distribution. The classifier outputs an image mask, highlighting the fungal infected region in the corneal image. The whole system is implemented in MATLAB.

Chapter 1

Introduction

For human beings, their eye provides a window to the world. It helps them sense the color, shape and state of physical objects. The Cornea, a part of the human eye, provides a physical barrier, and protects the inside of the eye from germs, dust and other harmful matter. Fungal Keratitis, a fungal infection caused by the fungus Aspergillus in the corneal layers, is one of the major causes of blindness if not diagnosed and treated at the earliest stages.

Fungi cannot penetrate an intact epithelium (the upper layer of Cornea) but can penetrate an injury or a previous epithelial defect and thereby enter the cornea. Once into the corneal layers, they are able to proliferate. To avoid permanent damage, it is important to diagnose and treat corneal infections at the earliest possible opportunity.

1.1 Fungal Infection Characterization Using Fractal Features and Statistical Pattern Classification

Because of the devastating ocular damage a fungal infection can produce, it becomes important to consider and experiment with techniques that can assist an accurate and timely diagnosis. Fractal features have been used for landscape classification, breast cancer diagnosis, (Low 1999, Priebe 1993, Priebe 1994) etc. In this work, an attempt has been made to classify the regions in the corneal images as fungal infected regions through statistical pattern classification technique using fractal features.

1.2 Statistical Pattern Classification

Much of the visual information we deal with during day to day activities involve detection of complex patterns such as faces, handwritten text, diseases, objects, etc. Though we still do not completely know the processes, which enable us to recognize these complex patterns, it has been widely accepted that a pattern must first be perceived by the sense organs before recognition (Fukunaga 1972). The advent of digital computers opened new avenues of research on ‘mechanization of though processes’ to enable the automation of monotonous repeated activities such as recognizing hand-printed text, sorting the mail, counting blood cells, analyzing cardiograms, recognizing spoken words, diagnosing diseases, etc.

Statistical pattern classification, a class of automatic pattern classification technique using statistics and probability theory, provides an adequate model for the variability of the pattern representations. The input to our pattern recognition machine is a set of d-dimensional vectors, the components of which are called the features. Feature measurements are acquired and converted into a form suitable for machine processing. The task of the pattern classifier is to assign the input to one out of ‘c’ possible classes. Two different patterns should be assigned to the same class if they are similar and to different classes if they are dissimilar. In pattern classification, features are chosen such that the feature distribution can uniquely identify the class of pattern.

1.3 Fractal Features For Fungal Infection Characterization

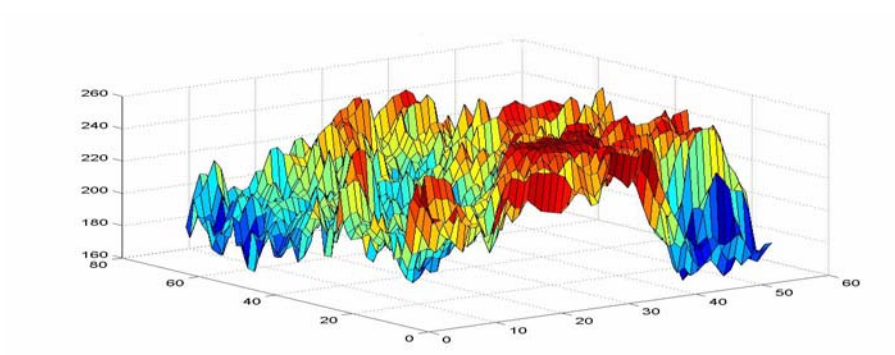

Figure 1. Gray-scale with fractal dimension between 2 & 3.

In a gray-scale image, each pixel can be assigned an integer value based on the pixel intensity. Hence a gray-scale image can be thought of as a rough surface embedded in a three-dimensional space as shown in the figure 1. From this viewpoint, the image can be considered to have a fractal dimension that is somewhere between the topological dimension of the image and the dimension of the space embedding the image i.e. a value between two and three. Thus, the fractal dimension can be viewed as a characterization of roughness, which can be used to classify the regions in the image using statistical pattern classification techniques. The problem of interest can be stated as follows: given a corneal image of an eye, which subsets of it can be identified as possessing fractal structure and for our purposes, can this fractal structure be used to classify portions of the image as fungal infected and non-fungal regions.

In chapter 2, we give a brief anatomy of the human eye and some pathological details about Fungal Keratitis.

In chapter 3, we give a brief introduction to fractals and estimation of fractal dimension using covering-blanket method. The Covering-blanket method estimates the fractal dimension based on the local computations via a windowing or averaging step, which is spatially dependent. These local estimates of fractal dimension will form a distribution and the distribution of computed fractal dimension may provide a better means of describing a texture than the mean fractal dimension of the texture.

In chapter 4, we explain the dimension reduction using Fisher linear discriminant (FLD).

Chapter 5 deals with estimating the probability density of the fractal features by the adaptive mixtures (AM) method.

In chapter7, we explain the Bayes classifier to classify the pixels of a new corneal image using the estimated probability density of the fractal features.

In chapter 8, we present our experimental results.

Chapter 9 concludes this work and gives suggestion for future work.

The entire work is implemented in MATLAB and the source code is presented in the appendix.

Chapter 2

Fungal Keratitis

2.1 Anatomy of the Eye

The human eye is the organ that gives us the sense of sight, allowing us to learn about the visual structure of our surrounding world than any of the other five senses. The eye allows us to see and interpret the shapes, colors, and dimensions of objects in the world by processing the light ray they reflect. The eye changes light rays into electrical signals then sends them to the brain, which interprets these electrical signals as visual images.

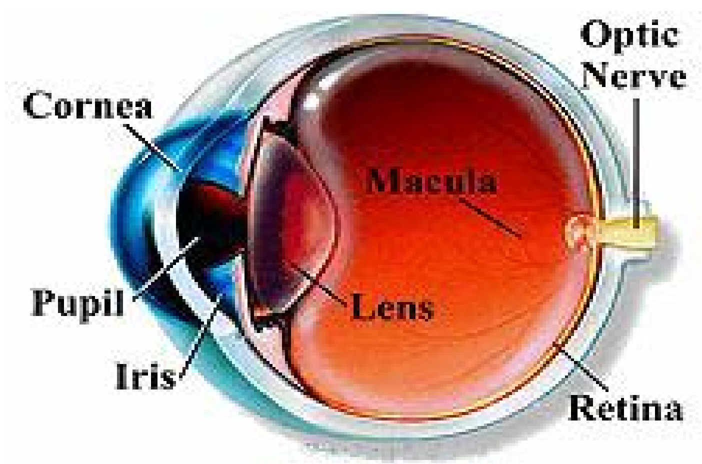

Figure 2. Human eye

The eye is set in a protective cone-shaped cavity in the skull called the orbit and measures approximately one inch in diameter (Crocco 1977). The orbit is surrounded by layers of soft, fatty tissue, which protect the eye and enable it to turn easily. Six muscles regulate the motion of the eye. Among the more important parts of the human eye are the iris, cornea, lens, retina, conjunctiva, the macula, and the optic nerve (Figure 2).

2.2 Cornea

The cornea is the front window of the eye. The cornea is as smooth and clear as glass but strong and durable as plastic; it helps the eye in two ways:

1. The cornea provides a physical barrier that shields the inside of the eye from germs, dust, and other harmful matter.
2. It acts as the eye's outermost lens. When light strikes the cornea, it bends or reflects the incoming light onto the crystalline lens. The lens then focuses the light onto the retina. Although the cornea is much thinner than the lens, the cornea provides about 65 percent of the eye's refractive power.

2.3 Importance of Cornea

As the eye's outermost tissue, the cornea functions like a window that controls the entry of light into the eye. The cornea filters out some of the most damaging ultraviolet (UV) wavelengths in sunlight. Without this protection, the lens and the retina would be highly susceptible to injury from UV radiation.

2.4 Structure of the Cornea

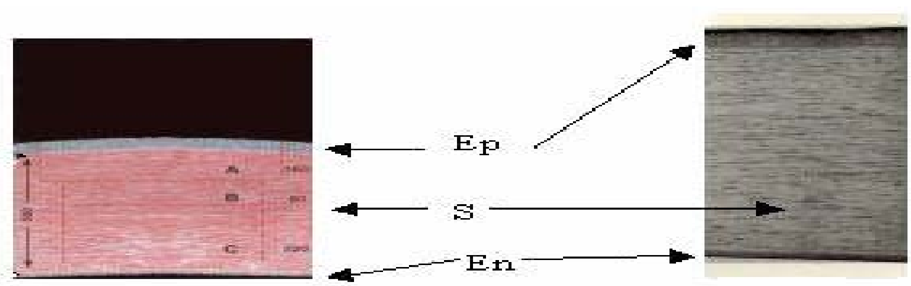

Figure 3. Structure of the Cornea
Ep – Epithelium; S – Stroma; En – Endothelium

Although the cornea is clear and seems to lack substance, it is actually a highly organized group of cells and protein. The cornea receives its nourishment from the tears and aqueous humor that fills the chamber behind it. Unlike most tissues in the body, the cornea contains no blood vessels to nourish and protect it against infection, which would disturb proper light refraction. The tissue is arranged in three main layers (Figure 3).

2.4.1 Epithelium

This is the outermost layer in the cornea, which comprises about 10 percent of the tissue's thickness. The Epithelium provides the following functions:

1. Block the passage of foreign material such as dust or water into the eye and other layers of the cornea.
2. Provide a smooth surface that absorbs oxygen and other needed cell nutrients that are contained in tears.

2.4.2 Stroma

The Stroma is located behind the epithelium and comprises about 90 percent of the cornea. It consists primarily of water, layered protein fibers that give the cornea its strength, elasticity, and form, and cells that nourish it. The unique shape, arrangement, and spacing of the protein fibers are essential in producing the cornea's light conducting transparency.

2.4.3 Endothelium

This single layer of cells is located between the stroma and the aqueous humor. Because the stroma tends to absorb water, the endothelium's primary task is to pump excess water out of the stroma. Without this pumping action, the stroma would swell with water, become hazy, and ultimately opaque.

2.5 Corneal Diseases

The cornea copes very well with minor injuries or abrasions. If dirt scratches the highly sensitive cornea, epithelial cells slide over quickly and patch the injury before an infection can occur. But if the scratch penetrates the cornea more deeply, the healing process will take longer, resulting in greater opportunity for invasion. Some of the most serious problems that affect the cornea are:

1. Microbial infections (Keratitis)
2. Conjunctivities ('Pink eye')
3. Ocular Herpes (Viral infection)
4. Corneal Dystrophies:
    i. Keratoconus
    ii. Map-Dot-Fingerprint Dystrophy
    iii. Fuch's Dystrophy
    iv. Lattice Dystrophy

In this work, an attempt is made to assist the diagnosis of 'Fungal Keratitis', a microbial infection in cornea, by identifying the region of infection in the gray scale image of the cornea (Stroma).

2.6 Fungal Keratitis

Leber (Leber 1879) first described fungal Keratitis in 1879. This entity is not a common cause of corneal infection, but it represents one of the major causes of infectious Keratitis in tropical areas of the world (See 1998). Aspergillus (Schmitt 1992), a fungus, is the most common cause of fungal Keratitis worldwide.

Chapter 3

Fractal Objects and Fractal Dimension Computation

In this chapter, we give a brief introduction to fractal objects and methods to estimate the fractal dimension. A Fractal is any object whose Hausdorff-Besicovitch dimension (Mandelbrot 1977) exceeds its topological dimension. The following are the characteristics of fractals:

1. Self-similar under all magnification (scales).
2. A measured property of a fractal can change with scale changes in a prescribed manner. Examples of measured properties are the line length of a one-dimensional signal and the surface area of a two-dimensional signal.

3.1 Measuring Fractal Curves

The incommensurability of the diagonal of a square was initially a problem of measuring length, which was resolved with irrational numbers. Attempts to compute the length of the circumference of the circle led to the discovery of the constant number π. But these long solved problems did not help us resolve some more recent measure problems. According to an encyclopedia in Spain, the length of the border between Spain and Portugal was 616 miles, while a Portuguese encyclopedia quotes 758 miles. Similarly, according to the various sources, the length of the coast of Britain vary anywhere between 4500 and 5000 miles. These differences are not due to measurement error. Rather, Mandelbrot's article 'How long is the coast of Britain?' demonstrated that typical coastlines do not have a meaningful length.

In practice we measure the coast on a geographical map of Britain rather than the real coast. For example, if the scale of the map is 1:1,000,000 and the compass width is

5cm, then the corresponding true distance is 5,000,000cm or 50km. Figure 4 shows these two separate polygonal representations of the coast of Britain.

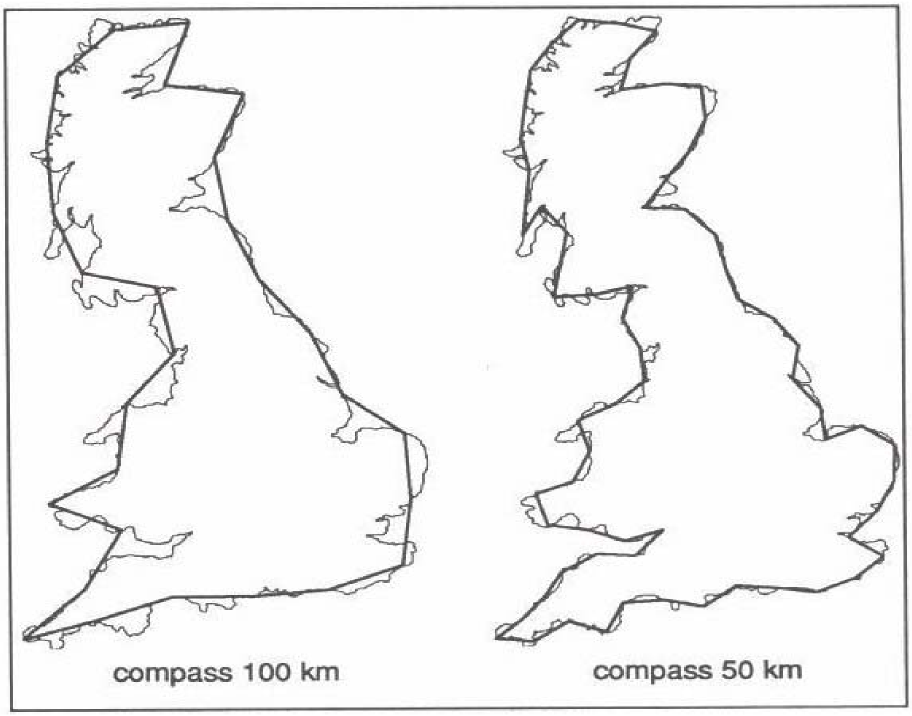

Figure 4. Polygonal approximation of the coast of Britain

The vertices of the polygons are assumed to be on the coast. The straight-line segments have constant length and represent the setting of the compass.

This experiment reveals a surprise. The smaller the setting of the compass, the more detailed the polygon is and the longer the resulting measurement will be.

The relationship between a measure property (length or area) and the measuring scale (ε) is defined by Richardson's law and is given by:

$M(\varepsilon) = K\varepsilon^{d-D}$

where,

ε is scale with values 1,2,..

$M(\varepsilon)$ is the value of the measured property at scale ε, D is the fractal dimension,

K is a constant,

d is the topological dimension.

3.2 Estimation of Fractal Dimension Using Covering Method

The fractal dimension can be estimated using various techniques. Some of the methods commonly used for computation of the fractal dimension are:

1. Box counting method
2. The Power-spectrum method
3. The difference-statistics method
4. And the covering-blanket method.

The Covering-blanket method (Peli 1990) is used in this work to compute the fractal dimension. The covering method views the image as a two manifold embedded in R^3 in order to estimate the surface area of the image in a window surrounding a pixel. An integer gray scale value g(i, j) is associated to each pixel (i,j). A sequence of upper and lower surfaces, U(i, j, ε) and L(i, j, ε) respectively, are defined, which bound the original image in gray scale intensity. For a scale value (ε) of 0 define

$$U(i,j,0) = L(i,j,0) = g(i,j)$$

For other scale values, U and L are defined as follows:

$$U(i,j,\varepsilon+1) = \max\left\{U(i,j,\varepsilon)+1, \max_{k;m\in\eta} U(k,m,\varepsilon)\right\}$$

$$L(i,j,\varepsilon+1) = \min\left\{L(i,j,\varepsilon)-1, \max_{k;m\in\eta} U(k,m,\varepsilon)\right\}$$

where,

$$\eta = \{(k,m) \mid \text{Euclidean distance}((i,j),(k,m)) \le 1\}$$

Using a window R centered at the pixel (i, j), an area estimate at that pixel is defined as follows:

$$A(i,j,\varepsilon) = (2\varepsilon)^{-1} \sum_{(k,l)\in R} \left[U(k,l,\varepsilon) - L(k,l,\varepsilon)\right]$$

Once this area estimate is obtained for several values of ε, by using a window R centered at each pixel, the fractal dimension D can be computed using Richardson's law. D is computed as the slope obtained via a regression of log[A(ε)] vs. log[ε] as shown in the figure 5.

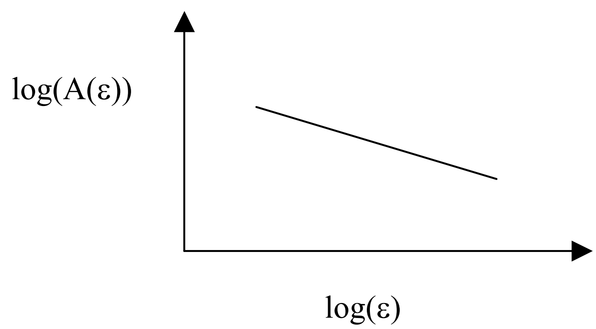

Figure 5. log(A(ε)) vs. log(ε) plot

The following three features can be extracted from the regression, for each pixel:

1. Slope (d – D)
2. Y intercept
3. F statistic (Goodness of fit)

The slope is directly related to the fractal dimension of the texture constructed using the window. The y-intercept is another naturally occurring property of the regression line. This represents a scale factor on the rate of area change with respect to ε. The F statistic represents the significance of the regression (Montgomery 1982).
Thus a 3D power law feature vector can be computed for each pixel in the image. In this work, a window R of size 7 x 7 is used. (Note that the value of the power law features

will vary based on the window size.) A method for computing a fractal dimension estimate, which is invariant with respect to the window size, is presented in (Low 1999). A more appropriate fractal dimension of the surface could be estimated by incorporating the segmentation boundaries as demonstrated in (Solka 1994).

In this thesis, these three features extracted from the regression plot, referred as the fractal features, will be used to classify the pixels to belong to the fungal infection region or non-fungal region.

Chapter 4

Curse of Dimensionality and Dimension Reduction

One of the recurring problems encountered in applying statistical techniques to pattern recognition problem is what Bellman calls the curse of dimensionality (Bellman 1961). Procedures that are analytically or computationally manageable in low-dimensional spaces can become intractable in a space of 50 or 100 dimensions. Thus various techniques have been developed for reducing the dimensionality of the feature space in the hope of obtaining a more computationally efficient problem.

The fractal features extracted from the grayscale image of the corneal layers of the eye can be thought of as co-ordinates of vectors in three-dimensional space. While more information is often contained in higher dimensional feature space, the difficulty associated with probability density function (pdf) estimation increases dramatically with any increase in the dimensionality of the observations (Scott 1992). The Fisher Linear Discriminant (FLD) (Duda 1973) can be used to simplify the computations that allow projecting the 3D raw features to 1D transformed features onto one dimension.

4.1 Fisher Linear Discriminant (FLD)

Fisher discriminants group features of the same class from the features of a different class. Features are projected from N-dimensional space (where N is the size of the feature vector) to C-1 dimensional space (where C is the number of classes of features). For example, consider two sets of points in 2-dimensional space that are projected onto a single line as shown in figure 6. Depending on the direction of the line, the points can either be mixed together (Figure 7) or separated (Figure 8). Fisher discriminants find the line that best separates the points.

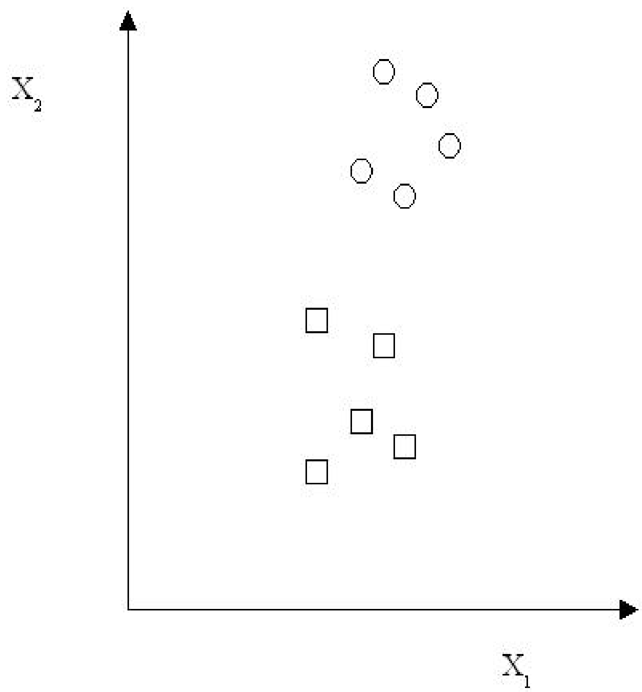

Figure 6. Two sets of points in a 2D space

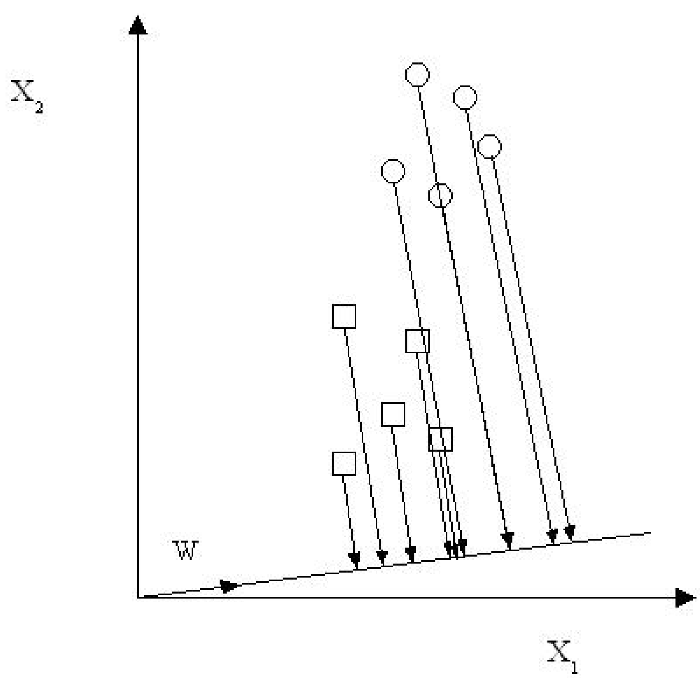

Figure 7. Two sets of points mixed together

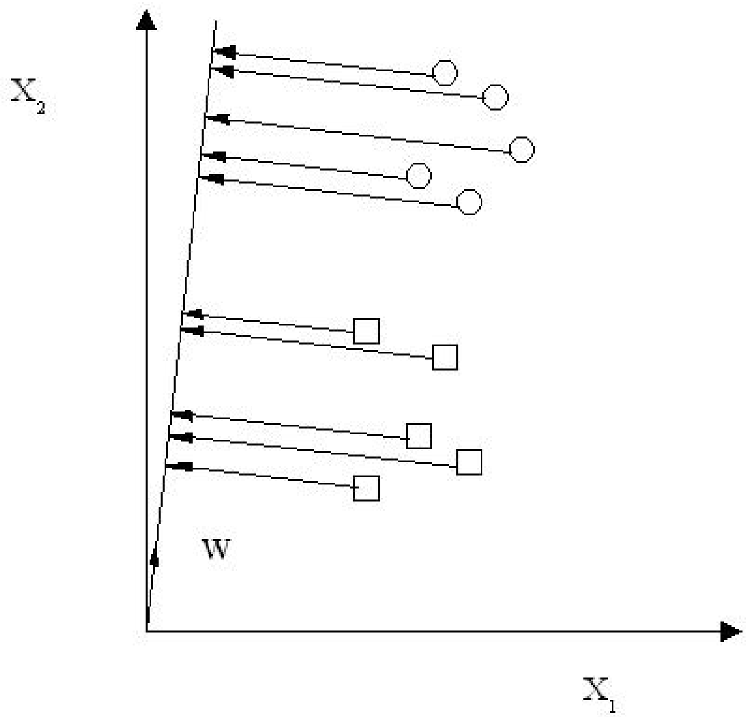

Figure 8. Two sets of points well separated

As with eigenspace projection, features extracted from the training images (training features), are projected into a subspace. The features extracted from the test images (test features), are projected into the same subspace for discriminant analysis. Following are the steps to follow to find the Fisher discriminants for a set of features.

- **Calculate the "within class" scatter matrix:**

The within class scatter matrix measures the amount of scatter between items in the same class. For the ith class, a scatter matrix (S_i) is calculated as the sum of the covariance matrices of the features in that class.

$$S_i = \sum_{x \in X_i} (x - m_i)(x - m_i)^T$$

where m_i is the mean of the features in that class. The within class scatter matrix (S_W) is the sum of all the scatter matrices.

$$S_w = \sum_{i=1}^{C} S_i$$

where C is the number of classes.

- **Calculate the "between class" scatter matrix:**

The between class scatter matrix (S_B) measures the amount of scatter between classes. It is calculated as the sum of the covariance matrices of the difference between the total mean and the mean of each class.

$$S_B = \sum_{i=1}^{C} n_i (m_i - m)(m_i - m)^T$$

where n_i is the number of features in the class, m_i is the mean of the features in the class and m is the mean of all the features.

- **Solve the generalized eigenvalue problem:**

Solve for the generalized eigenvectors (V) and eigenvalues (Λ) of the within class and between class scatter matrices.

$$S_B V = \Lambda S_W V$$

- **Keep the first C-1 eigenvectors:**

Sort the eigenvectors by their associated eigenvalues from high to low and keep the largest C-1 eigenvectors. These eigenvectors form the Fisher basis vectors.

- **Project the training features onto Fisher basis vector:**

Project all the training features onto the Fisher basis vector (w) by calculating the dot product of the features with the Fisher basis vector.

For each class of training data (the fungal and non-fungal), X_i represents the observations from class i (where i=1, 2 for two classes). Each observation vector $x=[x_1, x_2, x_3]^T$ formed with the fractal features, is contained in the set X_i. Each three-dimensional fractal feature vector x, is transformed into a one-dimensional vector y using the Fisher basis vector (w).

$y = w^T x$

where, $x \in X_i$

$y \in Y_i$

Now, the pdfs for Y_1 and Y_2, the set of transformed fractal features of the fungal and non-fungal regions of the training images, can be estimated using the adaptive mixtures technique (Chapter 5).

Chapter 5

Density Estimation Using Adaptive Mixtures (AM) Method

The probability density function (pdf) is a fundamental concept in statistics. Consider any random quantity X that has probability density function f. The probability density function f gives a natural description about the distribution of X, and allows probabilities associated with X to be found from the relation

$$P(a < X < b) = \int_a^b f(x)dx \text{ for all a < b.}$$

Suppose, that the observed data points are assumed to be a set of samples from an unknown probability density function. Density estimation is the construction of an estimate of the density function from the observed data. The following are two density estimation methods.

5.1 Parametric Density Estimation

This assumes the data to be drawn from one of a known parametric family of distributions, for example the normal distribution with mean μ and variance σ^2. The density f, underlying the data could then be estimated by finding the estimates of μ and σ^2 from the data.

5.2 Non-Parametric Density Estimation

This class of density estimation techniques uses the data to estimate the pdf. Here, the probability density will be a function of the fractal features of the grayscale corneal image of human eye. To estimate the pdf, the adaptive mixtures method described in (Priebe 1991) is used. The Adaptive mixtures (AM) method is a non-parametric density estimation method used to calculate the pdf without making strict assumptions about the actual statistical distribution of the data. It is a hybrid approach designed to maintain the best features of both kernel estimation and finite mixture models (Silverman 1986).

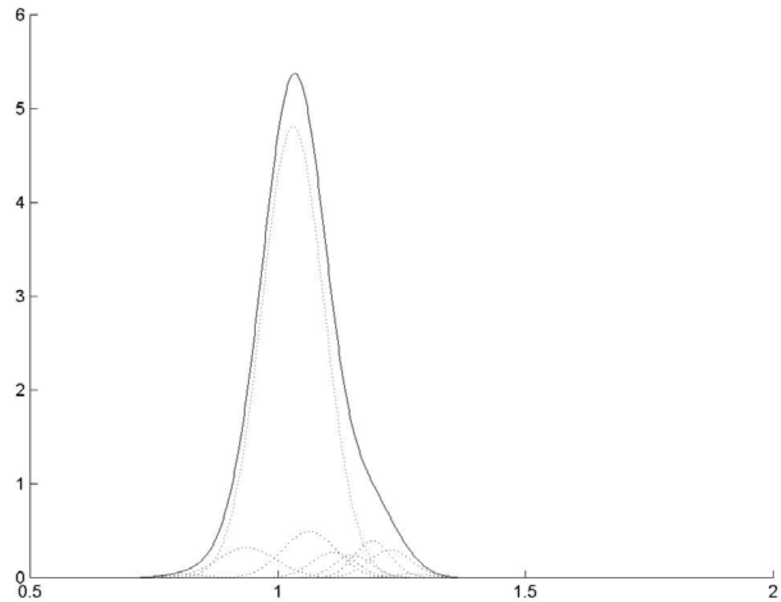

Figure 9. Density as a mixture of normal densities

The AM algorithm is a modification of a standard statistical technique called the method of mixtures, which represents the density as a mixture of normal densities (Figure 9). This method uses a data-driven approach for estimating the number of component densities in a mixture model based on the recursive Expectation/Maximization (EM) (Martinez 2002) algorithm. The basic idea behind the adaptive mixtures method is to take one data point at a time and determine the distance from the observation to each component density in the model. If the distance to each component is larger than some threshold, then a new term is created. If the distance is less than the threshold for all terms, then the parameter estimates are updated based on the recursive EM equations. The mixture density is given by

$$f(x) = \sum_{i=1}^{m} \pi_i \Phi(x; \mu_i, \sigma_i)$$

where,

π_i is the proportion of the i^{th} component,

μ_i is the mean of the i^{th} component,

σ_i is the standard deviation of the i^{th} component, and

$\Phi(x; \mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\left[\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right]}$, density of the i^{th} component.

Chapter 6

Building the Two Class Fractal Feature Distribution for Pixel Classification

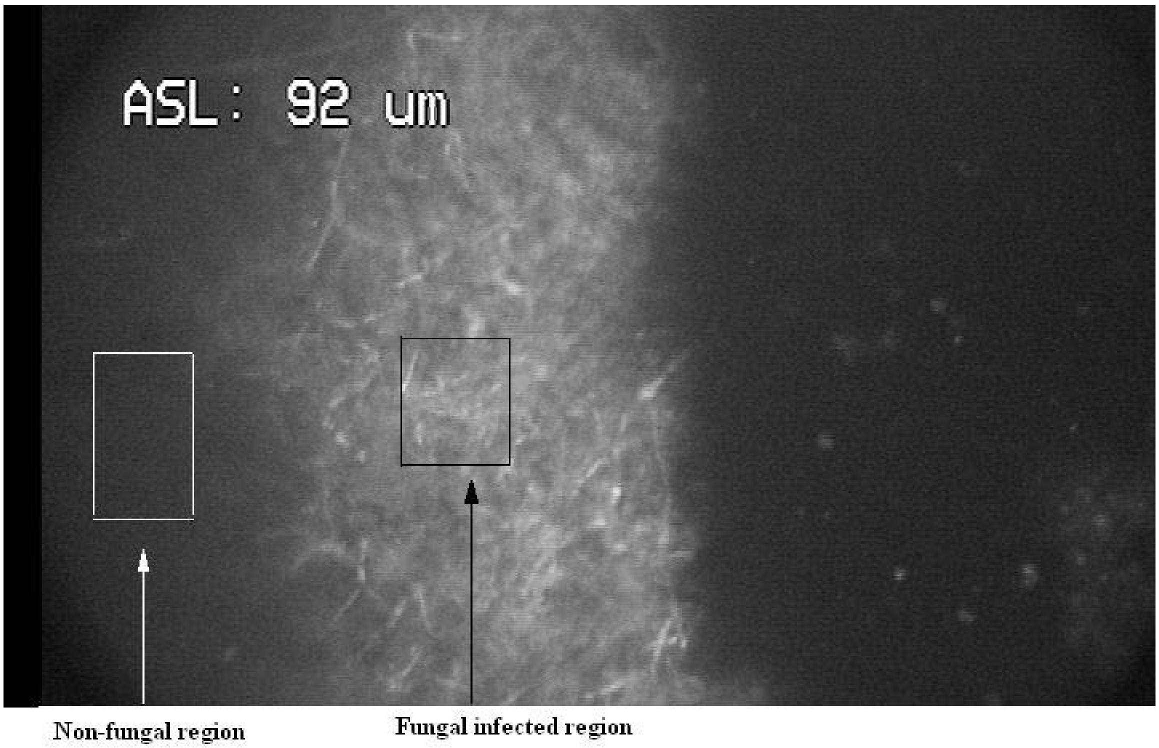

Figure 10. Corneal image of an eye with fungal infection

Images used in this study were provided by LSU Eye center, New Orleans. Using the covering-blanket method described in chapter 3, fractal features of the fungal and non-fungal regions in the training corneal image (Figure 10), are extracted. This forms the training data. A Fisher basis vector is obtained as explained under chapter 4, to transform the fractal features from three-dimension to one-dimension. Further, the pdfs for the transformed fractal features (one-dimension) are estimated using the adaptive mixtures method. Table 1 contains the proportions and density parameters of the mixture densities estimated using the training fractal features. Figure 11 shows the pdf plot of the fractal feature distribution of the fungal and non-fungal regions. The discriminant boundary is clearly evident from the pdf plot. Using this discriminant boundary, the pixels from the new corneal images can be classified.

Table 1. Component density parameters

	Pdf (Class1)			**Pdf (Class2)**		
Components	**Proportion (π_i)**	**Mean (μ_i)**	**Variance (σ_i^2)**	**Proportion (π_i)**	**Mean (μ_i)**	**Variance (σ_i^2)**
1.	0.0015	0.6413	0.0011	0.0034	0.8129	0.0019
2.	0.2192	0.8396	0.0038	0.0480	0.9374	0.0035
3.	0.6906	0.8660	0.0017	0.7564	1.0306	0.0039
4.	0.0101	0.8146	0.0014	0.0668	1.0632	0.0029
5.	0.0223	0.8404	0.0008	0.0303	1.1129	0.0020
6.	0.007	0.6757	0.0010	0.0227	1.1514	0.0014
7.	0.0226	0.8595	0.0004	0.0365	1.1907	0.0014
8.	0.0202	0.8692	0.0004	0.0322	1.2273	0.0019
9.	0.0114	0.8793	0.0009	0.0001	0.3551	0.0463
10.	0.0013	0.7241	0.0027	0.0039	1.2685	0.0015

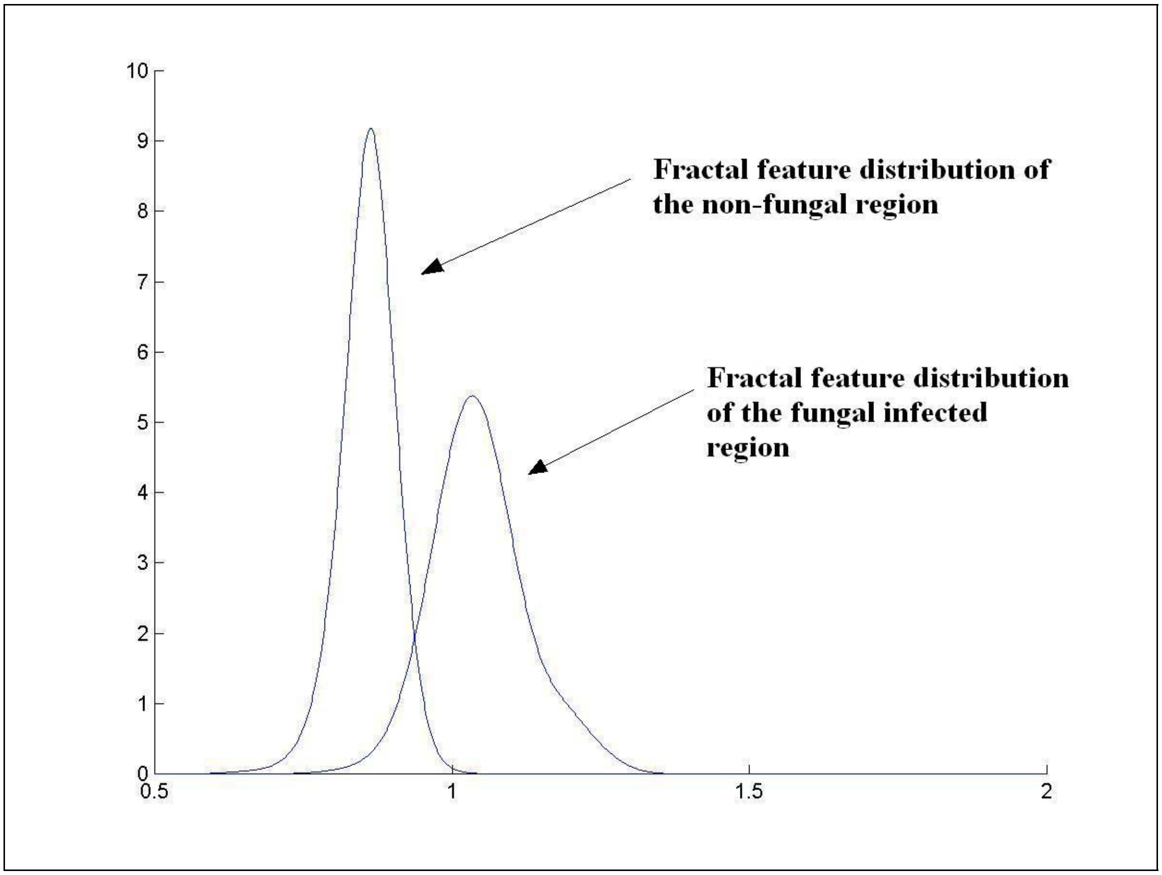

Figure 11. Fisher linear discriminant pdfs for the training corneal image

Chapter 7

Classifier

Bayes decision theory (Duda 1973) is a fundamental statistical approach to the problem of pattern classification. This approach is based on the assumption that the decision problem is posed in probabilistic terms.

Given a gray-scale corneal image, we need to classify the pixels into two classes: 1. Class1 (ω_1) – belonging to fungal infected region and 2. Class2 (ω_2) – belonging to the non-fungal region. Each pixel in the image belongs to either class1 (ω_1) or class2 (ω_2). Let ω denote the class each pixel belongs to, with $\omega = \omega_1$ for the class of pixels in the fungal infected region and $\omega = \omega_2$ for the class of pixels in the non-fungal region. (ω is considered to be a random variable).

Let, P(ω_1) and P(ω_2) represent the *a priori probabilities* for a pixel to belong to class1 and class2 respectively. These *a priori probabilities* reflect the prior knowledge about each class of pixels. P(ω_1) and P(ω_2) are non-negative and sum to one.

Let y be the 1D fractal feature of each pixel (derived as explained in chapter 4), which can be used as additional evidence to support the decision of pixel classification. Different samples of pixels will yield a different fractal feature value, and its variability can be expressed in probabilistic terms; let us consider y to be a continuous random variable whose distribution depends on the class of pixel (fungal and non-fungal).

Let, p(y | ω_j) be the state-conditional pdf of y, the pdf for y given that the pixel belongs to class ω_j . Figure 11 shows the pdf for p(y | ω_1) and p(y | ω_2). Suppose the *a priori probabilities* P(ω_j) and conditional densities p(y | ω_j) are known, the probability for any pixel having a feature value as likely as y to belong to class ω_j is given by the Bayes rule:

$$p(\omega_j \mid y) = \frac{p(y \mid \omega_j)P(\omega_j)}{p(y)}$$

where,

P(ω_j | y) – *a posteriori probability*

P(ω_j) – *a priori probability*

Now, having determined the probability for any pixel to belong to class ω_j given a fractal feature value of y, each pixel in a test image can be classified as follows:

1. Determine the 3D fractal feature vector for each pixel in the image (as explained in chapter 3).
2. Project all the 3D fractal feature vectors onto the Fisher basis vector w, obtained during the training phase. This would yield a 1D fractal feature vector y.
3. Using the conditional class densities p(y | ω_j) estimated under section, the *a posteriori probabilities* p(ω_1 | y) and p(ω_2 | y) can be estimated using the Bayes rule.
4. Now classify the pixel to belong to class ω_1 if

   P(ω_1 | y) > P(ω_2 | y)

   Otherwise, classify as class ω_2.

   Consider, P(ω_1 | y) > P(ω_2 | y).

   Using Bayes rule,

$\frac{p(y \mid \omega_1).P(\omega_1)}{p(y)} > \frac{p(y \mid \omega_2).P(\omega_2)}{p(y)}$, where p(y) is unimportant in decision-making.

$\therefore \frac{p(y \mid \omega_1)}{p(y \mid \omega_2)} > \frac{P(\omega_2)}{P(\omega_1)} = \mathrm{T}$

where,

P(ω_j) represents the *a priori probability* for any pixel to belong to class ω_j .

T is a threshold value.

This is referred as the likelihood ratio approach for hypothesis testing (Martinez 2002).

Since there is no prior knowledge about the probability of each class (fungal and non-fungal) of pixels, an equal *a priori probability* values are assumed. Hence a threshold value of 1 is assumed. And the pixel classification can be rewritten as follows:

If $\frac{p(y \mid \omega_1)}{p(y \mid \omega_2)} > 1$

Classify the pixel as class1.

Otherwise

Classify the pixel as class2

End.

Chapter 8

Results

Having obtained the Fisher basis vector and the two-class fractal feature distribution, any new corneal image can be processed. First, the 3D fractal feature vector is extracted from the new corneal test image using the covering method. Then, the 3D fractal feature vector is transformed to the 1D feature space using the Fisher basis vector (w). Now the pixels in the test image can be classified using the Bayesian classifier. Figures 12-21 show the set of test images and their corresponding mask highlighting the fungal region.

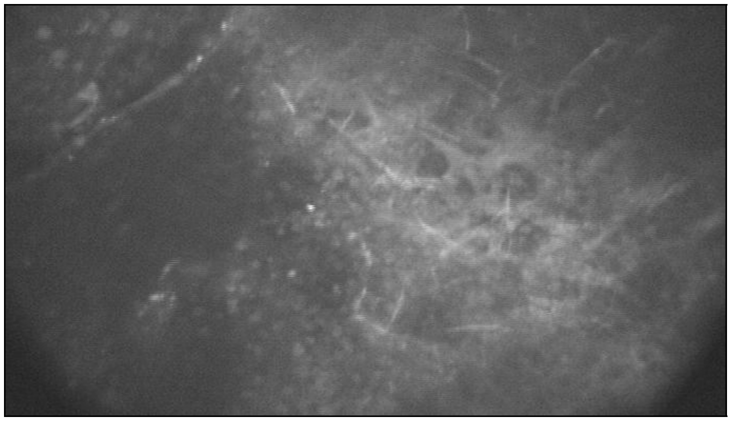

Figure 12. Test Image –1

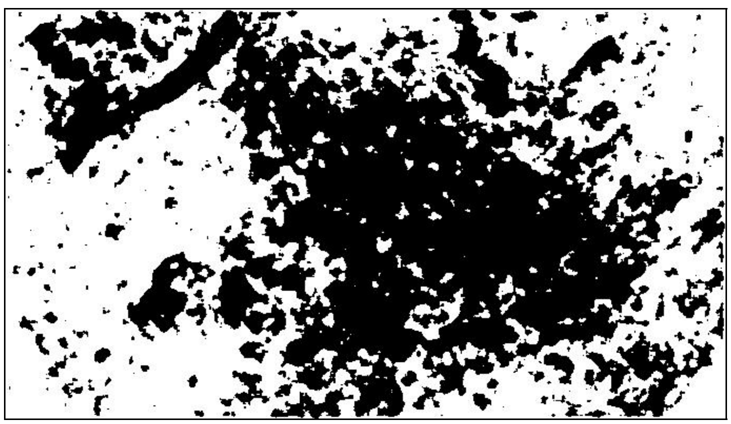

Figure 13. Mask - 1

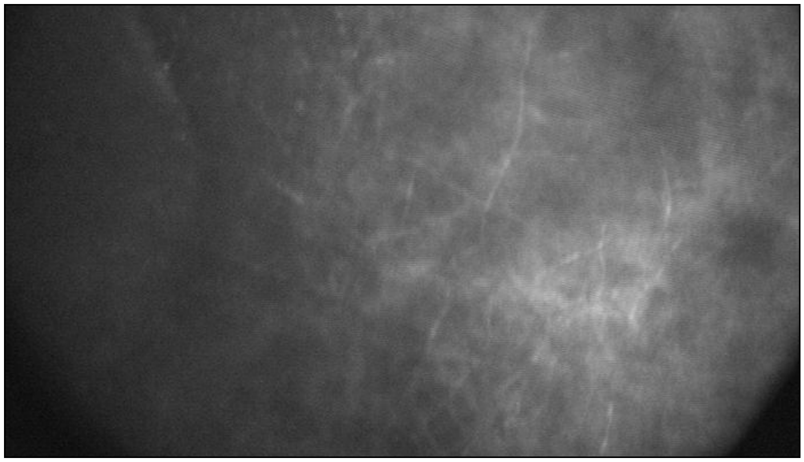

Figure 14. Test image – 2

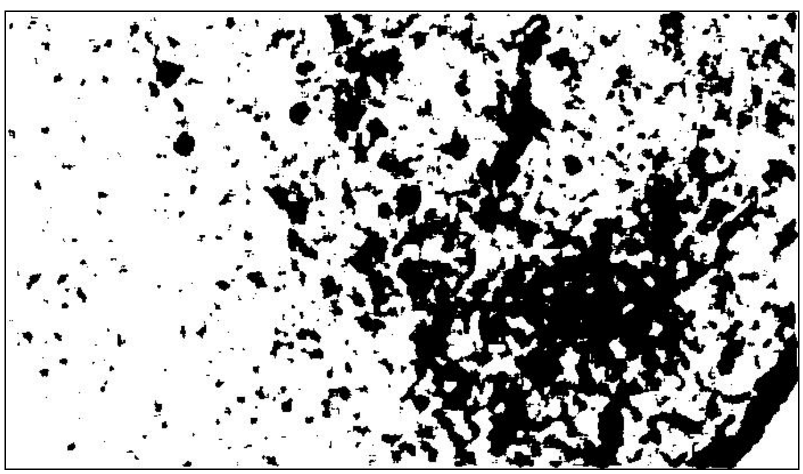

Figure 15. Mask – 2

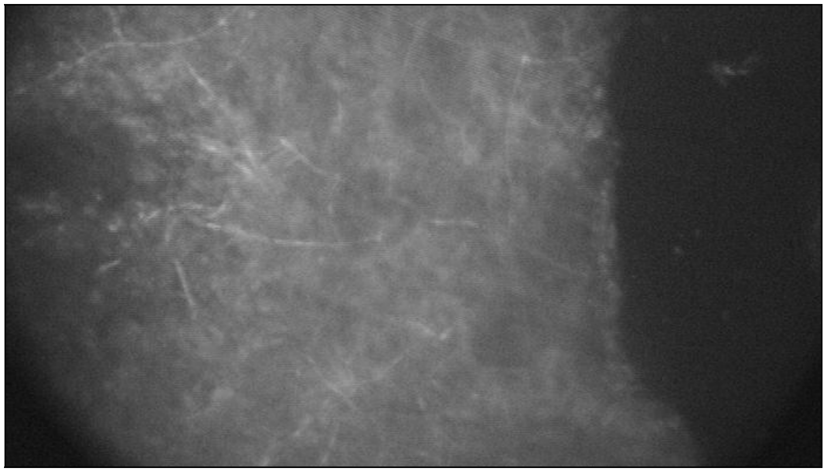

Figure 16. Test image – 3

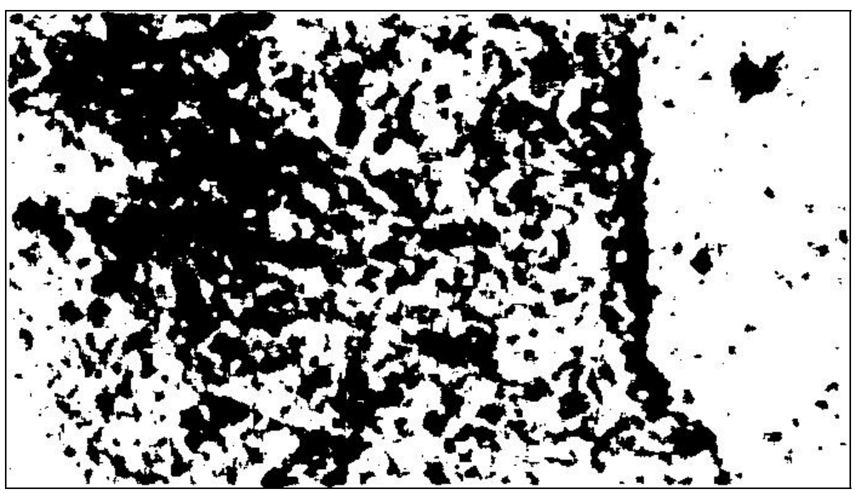

Figure 17. Mask – 3

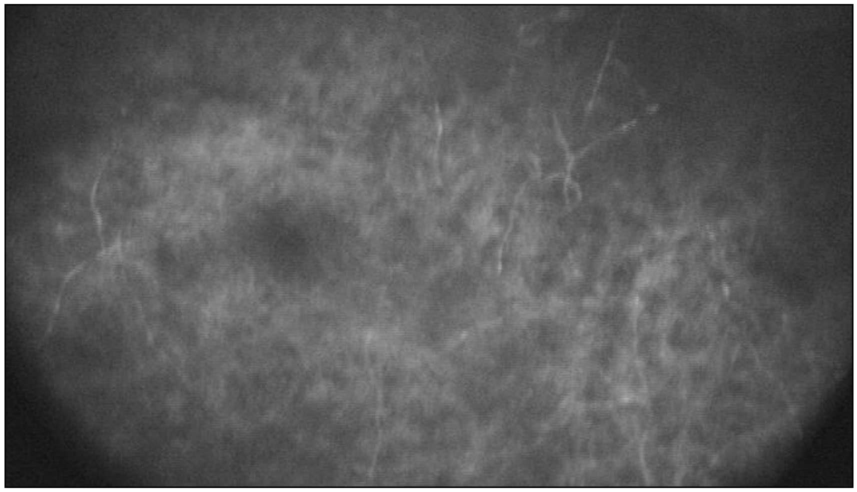

Figure 18. Test image – 4

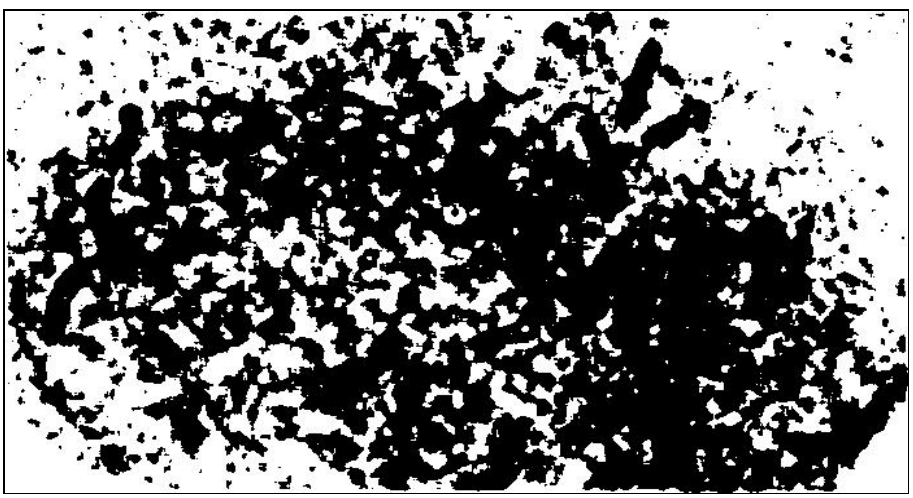

Figure 19. Mask – 4

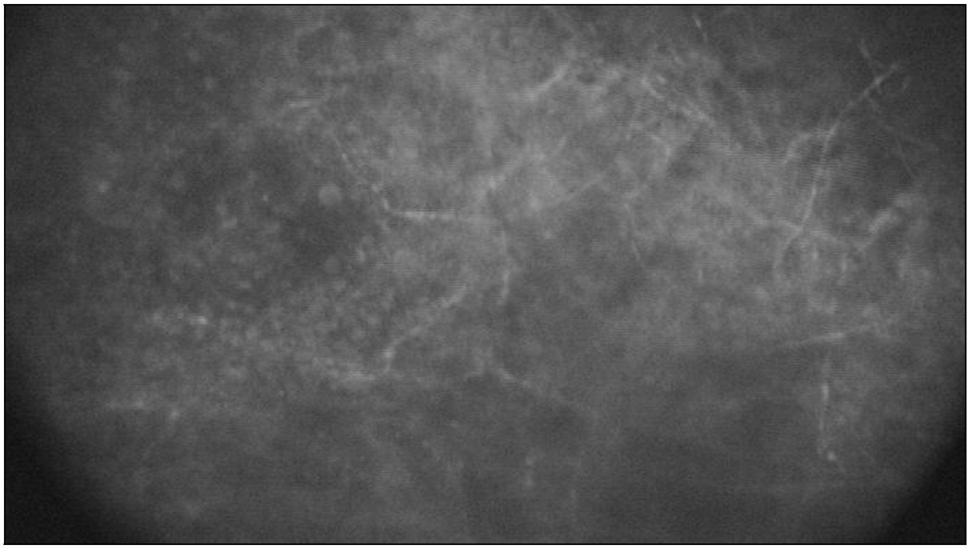

Figure 20. Test image – 5

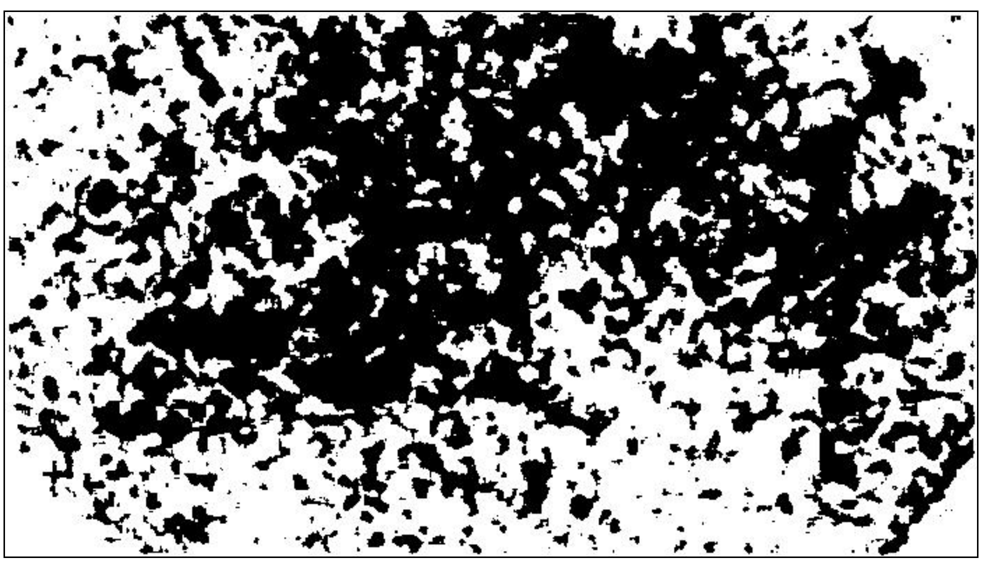

Figure 21. Mask – 5

Chapter 9

Conclusion and Future Work

Corneal diseases are one of the major causes of blindness and visual impairment. In this work, an attempt is made to assist the diagnosis of Fungal Keratitis, a fungal infection in the corneal layers, by identifying the region of infection using statistical pattern classification technique. Fractal dimension, a measure of surface roughness, is used to classify the regions in the image. It was found that the fungus infected regions and the non-fungal regions in the image exhibit a unique distribution of fractal features, which is used to classify the regions in new test images. It has been observed that fractal features are viable measures to classify the regions in corneal images. Also, when new patterns become available, the system can easily be trained by re-estimating the pdfs of the fractal features.

Fractal dimension computation algorithm is processor intensive and the fractal features are computed for each pixel in the image. Hence implementing the system on parallel computer cluster would help in real-time corneal image analysis for fungal infection diagnosis.

Bibliography

(Martinez 2002). Wendy L. Martinez, Angel R. Martinez, Computational statistics handbook with MATLAB, Chapman & Hall/CRC, 2002.

(Low 1999). H. K. Low, H. T. Chuah and H. T. Ewe, “A neural network landuse classifier for SAR images using textural and fractal information”, Geocarto International, Vol. 14, No. 1, March 1999.

(See 1998). J. L. S See, T. Y. Wong, K. T. Yeo, “Trends in the pattern of blindness and major ocular diseases in Singapore and Asia”, Annals Academy of Medicine Singapore, 27, pp. 540 – 546, 1998.

(Solka 1994). J. L. Solka, and D. J. Marchette, “Incorporating segmentation boundaries into the calculation of fractal dimension features”, Proceedings of the 26th symposium on the interface, Research Triangle park, NC, 1994.

(Priebe 1994). C. E. Priebe, J. L. Solka, R. A. Lorey, G. W. Rogers, W. L. Poston, M. Kallergi, W. Qian, L. P. Clarke, R. A. Clark, “The Application of Fractal analysis to Mammographic Tissue Classification”, Cancer Letters, 77, 183-189, 1994.

(Priebe 1993). C. E. Priebe, J. L. Solka, G. W. Rogers, “Discriminant analysis in aerial images using fractal based features”, Proceedings SPIE, Vol. 1962, pp. 196 – 208, 1993.

(Solka 1992). Jeffrey L. Solka, Carey E. Priebe, and George W. Rogers, “An initial assessment of discriminant surface complexity for power law features”, Simulation, 58, pp. 311 – 318, May 1992.

(Scott 1992). D. W. Scott, Multivariate Density Estimation, John Wiley and Sons, New York, NY, 1992.

(Schmitt 1992). H. J. Schmitt, F. Edwards, J. Andrade, Y. Niki and D. Armstrong, “Comparison of azoles against aspergilli in vitro and in an experimental model of pulmonary aspergillosis”, Chemotheraphy, 38, 118-126.

(Priebe 1991). C. E. Priebe, and D. J. Marchette, “Adaptive Mixture: recursive nonparametric pattern recognition”, Pattern Recognition, 24, pp. 1197-1209, 1991.

(Peli 1990). T. Peli, “Multiscale fractal theory and object characterization”, Journal of optical society of America, Vol. 7, No. 6, June 1990.

(Silverman 1986). B. W. Silverman, Density Estimation, Chapman and Hall, New York, 1986.

(Montgomery 1982). D. C. Montgomery, and E. A. Peck, Introduction to Linear Regression Analysis, New York, John Wiley and Sons, 1982.

(Mandelbrot 1977). Benoit B. Mandelbrot, The fractal geometry of nature, W. H. Freeman and company, New York, 1977.

(Crocco 1977). John A. Crocco, The classic collector's edition GRAY'S ANATOMY, Gramercy Books, New York, 1977.

(Duda 1973). Richard O. Duda, Peter E. Hart, Pattern classification and scene analysis, John Wiley and Sons, New York, 1973.

(Fukunaga 1972). Keinosuke Fukunaga, Introduction to statistical pattern recognition, Academic press, Inc., London, 1972.

(Bellman 1961). R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.

(Leber 1879). T. Leber, "Keratomycosis asperfillina als Ursache von Hypopyonkeratitis", Albrecht von Graefes Archiv fůr Ophthalmolgie, 25, 285-301, 1879.

Appendix

MATLAB Source Code

function Diagnosis

```
% Training phase:
% ===============
% The training images: fimg1,2,3,4,wimg1,2,3 & 4.
% Fractal features are extracted from the images.
% Two class feature distribution built from the R3->R1 transformed feature.
% The classification boundary can be infered from the density distribution plot.

temp1 = fractalfeature('c:\Fungus\fimg1.tif');

temp2 = fractalfeature('c:\Fungus\fimg2.tif');
temp = cat(2,temp1,temp2);

% Feature vector (3D) for fungal infected image
C1 = temp;


% Extract the feature vector, for the healthy tissue in the eye
clear temp temp1 temp2;

temp1 = fractalfeature('c:\Fungus\wimg1.tif');

temp2 = fractalfeature('c:\Fungus\wimg2.tif');
temp = cat(2,temp1,temp2);

% Feature vector (3D) for fungal infected image
C2 = temp;
clear temp temp1 temp2;

%Find a discriminant boundary for the two class features
[w,ccc1,ccc2] = DimensionReduction(C1, C2);

%Save the intermediate results
save ccc1 ccc1;
save ccc2 ccc2;

save w w;

%Find the mixture densities Phi1 and phi2 of the two class data;
%Plot the derived densities
[pihatx, muhatx, varhatx, pihaty, muhaty, varhaty] = AdaptiveMixture(ccc1, ccc2);
```

```
save pihatx pihatx;
save muhatx muhatx;
save varhatx varhatx;
save pihaty pihaty;
save muhaty muhaty;
save varhaty varhaty;
```

function testImage(inImage, outImage, pihatx, muhatx, varhatx, pihaty, muhaty, varhaty)

```
% To classify the pixels in the test image 'inImage' using the two class training features
% Syntax:
%       testImage(inImage, outImage, pihatx, muhatx, varhatx, pihaty, muhaty, varhaty)
%-------------------------------------------------------------------------------------------

disp('Fractal Feature extraction')
C = fractalfeature(inImage);

load w;

disp('Transform R3 to R1')
ccc = Project(w, C);

% Density estimates alpha1 and alpha2 for the test image features
disp('Classify the pixels in the image');
img = imread(inImage, 'tif');

[m,n]=size(img);
cccIndex = 1
for i = 1:n,
  for j = 1:m,
    [alpha1] = DensityGivenX(ccc(1, cccIndex), pihatx, muhatx, varhatx);
    [alpha2] = DensityGivenX(ccc(1, cccIndex), pihaty, muhaty, varhaty);

    %Reset the pixel intensity
    img(j,i) = 0;

    if alpha1 >= alpha2,
      img(j,i) = 255;
    end;
    cccIndex = cccIndex+1;
  end;
end;

imwrite(img, outImage,'tif');
```

function [alpha] = DensityGivenX(x, pis, mus, vars)

```
% Returns the density estimate for a given X-value.
% Syntax:
%       [alpha] = DensityGivenX(x, pis, mus, vars)
% -------------------------------------------------

nterm = length(pis);

alpha = 0.0;
for i = 1 : nterm,
   alpha = alpha + pis(i)*exp(-0.5*(x - mus(i)).^2/vars(i))/((2*pi*vars(i)).^0.5);
end;
```

```
function [w, ccc1, ccc2] = DimensionReduction(C1, C2);

% Projects R3 feature to R1 feature space using fisher's linear discrimant analysis
% Finds the eigen basis for the two class classification (w)
% Projects the two class data (ccc1, ccc2) using the newly found eigen basis (w)
% Syntax:
%       [w, ccc1, ccc2] = DimensionReduction(C1, C2);
%

[t1, len1] = size(C1);

[t1, len2] = size(C2);

C = cat(2, C1, C2);
[t1, len] = size(C);

clear t1;
ObsPerClass(1) = len1;
ObsPerClass(2) = len2;

NoClasses = 2;
NoFeatures = 3;

%Compute mean of each class
mn = mean(C')';

m(:,1) = mean(C(:,1:ObsPerClass(1))')';
m(:,2) = mean(C(:,ObsPerClass(1)+1:len)')';

%Compute between-class scatter matrix
%msm(:,1) = m(:,1) - mn;
%msm(:,2) = m(:,2) - mn;

%sb = zeros(NoFeatures);
%for i = 1:NoClasses,
%   sb = sb + msm(:,i) * msm(:,i)';
%end;

msm = m(:,1) - m(:,2);

sb = zeros(NoFeatures);
sb = msm * msm';

%Compute within-class scatter matrix
for i = 1:ObsPerClass(1),
   msc(:,i) = C(:,i) - m(:,1);
```

```
end;

for i = ObsPerClass(1)+1 : len,
   msc(:,i) = C(:,i) - m(:,2);
end;

sw = zeros(NoFeatures);
for i = 1:len,
   sw = sw + msc(:,i) * msc(:,i)';
end;

%Solve the generalized eigenvalue problem sb*w = d*sw*w
[V,D] = eig(sb,sw);
szd = size(D);
for i = 1:szd(1)
   evals(i) = D(i,i);
end;
[a, b] = sort(evals);

%Pull off the eigenvectors associated with the c-1 most significant eigenvalues
for i = 1 : NoClasses - 1
   w(:, i) = V(:, b(szd(1) - (i - 1)));
end;

%check that eig worked
norm((sb * w) - (D(b(szd(1)), b(szd(1)))*sw*w));

%Project two classes onto eigenvectors
for i = 1:len,
   ccc(:,i) = real(w' * C(:,i));
end;

%Form the projected 1D feature vectors (two class)
ccc1 = ccc(1:ObsPerClass(1));
ccc2 = ccc(ObsPerClass(1)+1:len);
```

function [ccc] = Project(w, C);

```
[x,len] = size(C);
%Project feature vector in the direction of the eigen basis w;
for i = 1:len,
   ccc(:,i) = real(w' * C(:,i));
end;

save ccc ccc;
```

function [C] = fractalfeature(Image)

```
%Computes the 3-dimensional feature vector
%Feature 1. Fractal dimension computed using covering method
%       2. F-ratio - goodness of fit for the regression line
%       3. Y-intercept of the regression line
%Syntax:
%       [C] = fractalfeature(Image)
%

A=imread(Image,'tif');

%number of epsilon values
no=8;

[row,col]=size(A);

ep=1;
for i=1:row,
 for j=1:col,
  u(i,j,ep)=double(A(i,j));
  l(i,j,ep)=double(A(i,j));
 end;
end;

%row=3;
%col=3;

%computing u1, l1 - upper and lower surfaces for the texture
for ep=2:no+1,
 for i=1:row,
  for j=1:col,
   u1_max=u(i,j,ep-1)+1;
   l1_min=l(i,j,ep-1)-1;

   if i-1 > 0
     u1_max=max(u(i-1,j,ep-1),u1_max);
     l1_min=min(l(i-1,j,ep-1),l1_min);
   end;

   if j-1 > 0
     u1_max=max(u(i,j-1,ep-1),u1_max);
     l1_min=min(l(i,j-1,ep-1),l1_min);
   end;

   if i+1 <= row,
```

```
    u1_max=max(u(i+1,j,ep-1),u1_max);
    l1_min=min(l(i+1,j,ep-1),l1_min);
   end;

   if j+1 <= col
    u1_max=max(u(i,j+1,ep-1),u1_max);
    l1_min=min(l(i,j+1,ep-1),l1_min);
   end;

   u(i,j,ep)=u1_max;
   l(i,j,ep)=l1_min;
  end;
 end;
end;

%computing the volume embedded between the surfaces with an observation window R
of 7X7 pixels
%volume contained in the observation window R, centered in the pixel (i,j) - v(i,j)
window=3;

for ep=2:no+1,
 for i=1:row,
  for j=1:col,
   %observation window centered at the pixel (i,j)
   %calculate the observation window co-ordinates

   %calculate x1 of (x1,y1);
   found=0;
   w=window;
   while w>0 & found==0,
    if i-w > 0
       x1=i-w;
       found=1;
       break;
    else
       w=w-1;
    end;
   end;

   if found ~= 1
     x1 = i;
   end;


   %calculate y1 of (x1,y1);
   found = 0;
```

```
w = window;
while w > 0 & found == 0,
  if j-w > 0
     y1 = j-w;
     found = 1;
     break;
  else
     w = w - 1;
  end;
end;

if found ~= 1
  y1 = j;
end;

%calculate x2 of (x2,y2);
found = 0;
w=window;    while w > 0 & found == 0,
  if i+w <= row
    x2 = i+w;
    found = 1;
    break;
  else
    w=w-1;
  end;
end;

if found ~= 1
  x2 = i;
end;

%calculate y2 of (x2,y2);
found = 0;
w=window;
while w>0 & found == 0,
  if j+w <= col
    y2 = j+w;
    found = 1;
    break;
  else
    w=w-1;
  end;
end;

if found ~= 1
  y2=j;
```

```
   end;

   %Got the window co-ordinates (x1,y1) and (x2,y2);

   %compute the volume under the pixel
   vol(i,j,ep)=0;
   for x=x1:x2,
    for y=y1:y2,
      vol(i,j,ep)=vol(i,j,ep)+u(x,y,ep)-l(x,y,ep);
    end;
   end;

  %end of j-loop
  end;
 %end of i-loop
 end;
%end of ep-loop
end;

clear x;
clear y;
clear u1_max;
clear l1_min;
clear i;
clear j;
clear x1;
clear y1;
clear x2;
clear y2;
clear found;
clear ans;
clear w;

%compute the area
%note: log(area) stored for computing fractal dimension.
for ep=2:no+1,
 for i=1:row,
  for j=1:col,
   area(i,j,ep)=log(vol(i,j,ep)/(2*ep));
  end;
 end;
end;

% log (ep)
for i=2:no+1,
 eplog(i) = log(i-1);
```

```
end;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% least square line - beta0, beta1: parameter estimation

% 1. Estimate beta0 and beta1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i = 1:row,
 for j = 1:col,
  SigXY = 0;
  SigX = 0;
  SigY = 0;
  SigXSq = 0;
  SqSigX = 0;
  Sxy = 0;
  Sxx = 0;

  for k = 2:no+1,
   SigXY = SigXY + eplog(k)*area(i,j,k);
   SigX = SigX + eplog(k);
   SigY = SigY + area(i,j,k);
   SigXSq = SigXSq + eplog(k)*eplog(k);
  end;

  SqSigX = SigX.^2;

  Sxy = SigXY - ((SigX * SigY) / no);
  Sxx = SigXSq - (SqSigX / no);

  beta1(i,j) = Sxy / Sxx;

  MeanX(i,j) = SigX / no;
  MeanY(i,j) = SigY / no;

  beta0(i,j) = MeanY(i,j) - beta1(i,j) * MeanX(i,j);

 end;
end;

%3. Find the F-ratio (f-statistic) - goodness of fit estimate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%3.1 Compute the estimated-area (log) using beta0 and beta1
clear i j k;
for i = 1:row,
  for j = 1:col,
    for k = 2:no+1,
      EstimatedArea(i,j,k) = beta0(i,j) + beta1(i,j) * eplog(k);
    end;
  end;
end;

for i = 1:row,
  for j = 1:col,
    SSR = 0;
    SSE = 0;

    for k = 2:no+1,
      SSR = SSR + (EstimatedArea(i,j,k) - MeanY(i,j)).^2;
      SSE = SSE + (area(i,j,k) - EstimatedArea(i,j,k)).^2;
    end;

    MSR = SSR / 1;
    MSE = SSE / (no - 2);

    Fratio(i,j) = MSR / MSE;
  end;
end;

%Compute the fraction dimension for each pixel
clear i j k;
for i = 1:row,
  for j = 1:col,
    fd(i,j) = 2 - beta1(i,j);
  end;
end;


%Feature vector [fd, beta0, Fratio]
TotElements = row*col;

f1 = reshape(beta1, TotElements, 1);
f2 = reshape(Fratio, TotElements, 1);
f3 = reshape(beta0, TotElements, 1);

for i=1:TotElements,
  C(1,i) = f1(i);
  C(2,i) = f2(i);
```

```
  C(3,i) = f3(i);
end;

%save C2 C;
```

Vita

Madhusudhanan Balasubramanian received his diploma in electrical and electronics engineering from Muthiah Polytechnic, Annamalainagar, India, 1995. He received his bachelor of engineering degree from Bharathiyar University, Coimbatore, India, 1998. He worked as a software engineer at IMR Global Ltd., Bangalore, India, and as a senior systems engineer at Wipro Technologies Ltd., Bangalore, India from June 1998 to July 2001. He joined the department of Computer Science at Louisiana State University (LSU) in fall 2001. He has attended various research workshops and has presented his research results in various research conferences. His areas of interest include, uncertainty analysis, data mining, bio-informatics, medical image processing and networking. He is currently pursuing his doctoral studies at LSU and will be receiving the degree of Master of Science in Systems Science, in December 2003.